Applied Soft Computing xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Iterated local search using an add and delete hyper-heuristic for university course timetabling

Jorge A. Soria-Alcaraz[a,*], Ender Özcan[b], Jerry Swan[c], Graham Kendall[b,d], Martin Carpio[e]

[a] Departamento de Estudios Organizacionales, División de Ciencias Economico Administrativas, Universidad de Guanajuato, Mexico
[b] University of Nottingham, School of Computer Science Jubilee Campus, Wollaton Road, Nottingham NG8 1BB, UK
[c] York Centre for Complex Systems Analysis, University of York, UK
[d] University of Nottingham Malaysia Campus, Jalan Broga, 43500 Semenyih, Selangor Darul Ehsan, Malaysia
[e] Tecnológico Nacional de México, Instituto Tecnológico de León, Mexico

ARTICLE INFO

Article history:
Received 10 December 2014
Received in revised form
16 November 2015
Accepted 28 November 2015
Available online xxx

Keywords:
Hyper-heuristic
Iterated local search
Add–delete list
Methodology of design
Educational timetabling

ABSTRACT

Hyper-heuristics are (meta-)heuristics that operate at a higher level to choose or generate a set of low-level (meta-)heuristics in an attempt of solve difficult optimization problems. Iterated local search (ILS) is a well-known approach for discrete optimization, combining perturbation and hill-climbing within an iterative framework. In this study, we introduce an ILS approach, strengthened by a hyper-heuristic which generates heuristics based on a fixed number of add and delete operations. The performance of the proposed hyper-heuristic is tested across two different problem domains using real world benchmark of course timetabling instances from the second International Timetabling Competition Tracks 2 and 3. The results show that mixing add and delete operations within an ILS framework yields an effective hyper-heuristic approach.

1. Introduction

Hyper-heuristics are (meta-)heuristics that choose or generate a set of low level (meta-)heuristics in an attempt to solve difficult search and optimization problems [1,2]. Heuristics can be used to search the solution space directly or construct a solution based on a sequence of moves. Hyper-heuristics aim to replace bespoke approaches by more general methodologies with the goal of reducing the expertise required to construct individual heuristics [3]. In most of the previous studies on hyper-heuristics, low-level heuristics are uniform, i.e. they are either constructive or perturbative (improvement) heuristics [4].

Educational timetabling problems are common and recurring real-world constraint optimization problems which are known to be NP-hard [5–7]. An educational timetabling problem requires scheduling of a set of events using limited resources subject to a set of constraints. There are a range of educational timetabling problems, such as examination timetabling and high school timetabling. This study focuses on the *university course time-tabling problem*, which can be further categorized as either *post-enrollment problems*, in which the student enrollment is available before the timetabling process, and *curriculum-based problems* in which the curricula of the students are known, but not the student enrollment [8]. There are two main types of constraints in a timetabling problem: *hard* and *soft* constraints. The hard constraints have to be satisfied in order to obtain a *feasible* solution, while violations of soft constraints are allowed, since they represent preferences. It is still the case at some universities that timetables are constructed by hand. Considering the inherent difficulty of generating high-quality feasible timetables which violate few soft constraints, it is usually desirable to automate timetable construction to improve upon solutions obtained by human experts [9]. However, automation of timetabling is not an easy task, since designing an automated method frequently requires a deep knowledge of the problem itself as well as the particular characteristics of the instance to be solved. This knowledge, in most cases, is not readily available to the typical researcher/end-user.

In this study, we describe an iterated local search (ILS) algorithm hybridized with a hyper-heuristic that generates heuristics based on add–delete operations to solve examination and university course timetabling problems. Re-usability, modularity and flexibility are some of the key features of the proposed approach. To evaluate the generality of the generation hyper-heuristic, it is tested on a range of problem instances across two different domains; namely, post-enrollment university course timetabling and curriculum-based university course timetabling, without modification of the underlying solution framework.

Although the problem domains we investigate are timetabling problems, each domain exhibits differing characteristics, particularly with respect to the complexity of the real-world constraints. This is the main reason why a recent competition has used two tracks. The International Timetabling Competition series was organized to create a common ground for the cross-fertilization of ideas, bridging the gap between theory and practice and creating a better understanding between researchers and practitioners in this field [8]. The second competition in the series (ITC2007) was on educational timetabling, containing an examination timetabling track and two separate tracks for post-enrollment and curriculum-based university course timetabling [8]. We have investigated the performance of the proposed approach on the last instances. The results show that our approach is promising.

* Corresponding author. Tel.: +52 4777751263.
E-mail addresses: jorge.soria@ugto.mx (J.A. Soria-Alcaraz), ender.ozcan@nottingham.ac.uk (E. Özcan), jerry.swan@york.ac.uk (J. Swan), graham.kendall@nottingham.edu.my (G. Kendall), jmcarpio61@hotmail.com (M. Carpio).

http://dx.doi.org/10.1016/j.asoc.2015.11.043
1568-4946/© 2015 Published by Elsevier B.V.

This paper is organized as follows. Section 2 provides an overview of educational timetabling problems, particularly university course timetabling. This section also discusses solution methodologies. Section 3 discusses the specifics of the solution methodology including the relevant data structures and the add–delete representation. Section 4 summarizes the experimental results. Finally, Section 5 presents the conclusions and future work.

2. Background

2.1. Hyper-heuristics

The term "hyper-heuristic" is relatively new, having first appeared in a technical report by Denzinger et al. [62] as a strategy to combine artificial intelligence methods. The un-hyphenated version of the term initially appeared in Cowling et al. [3] describing hyper-heuristics as *heuristics to choose heuristics* in the context of combinatorial optimization. However, the idea of automating the design of heuristic methods is not new and can be traced back to the 1960s in works such as Fisher et al. [11] and Crowston et al. [12].

The main motivation behind hyper-heuristic research is to reduce the need for human experts in designing effective algorithms, and consequently to raise the level of generality at which search methodologies are able to operate. Hyper-heuristics share the quest for greater autonomy and generality with approaches such as autonomous search by Hamadi et al. [13], reactive search by Battiti [14], adaptive operator selection by Maturana et al. [15], adaptive memetic algorithms [16], automated tuning [17] and parameter control by Lobo et al. [18]. In a recent book chapter by Burke et al. [4], the authors extended the definition of hyper-heuristics and provided a unified classification which captures more recent work that is being undertaken in this field. A hyper-heuristic is defined as a "*search method or learning mechanism for selecting or generating heuristics to solve computational search problems*". The classification of approaches considers two dimensions: (i) the nature of the heuristics' search space, and (ii) the different sources of feedback information from the search space. According to the nature of search space, we have;

- *Heuristic selection*: methodologies for choosing or selecting existing heuristics.
- *Heuristic generation*: methodologies for generating new heuristics from given components.

Orthogonal to the notion of selective versus generative is the distinction between constructive and perturbative mechanisms for searching the solution space, i.e. whether it operates via partial or complete solutions respectively.

This study describes an ILS which uses a generative hyper-heuristic for creating perturbation heuristics (move operators). The important feature of the proposed approach is the use of an add–delete list (i.e. a sequence of insertions or deletions of partial solution states) which acts like a ruin-recreate operator as proposed by Swan et al. [19]. This idea of removing and reinserting parts of the solution has produced encouraging results in previous work, for example: Schrimpf et al. for Vehicle Routing [20] and Misevicius et al. [21,22] for Quadratic Assignment. It is important to note that not all add–delete lists are feasible. In Section 3.2.1, we describe a divide-and-conquer algorithm for building a feasible add–delete list.

Fig. 1 illustrates the traditional framework for selective hyper-heuristics, with the *domain barrier* insulating the high-level search strategy from the underlying problem domain. The high-level strategy selects and applies a low-level heuristic (move operator) from the available set considering only (the history of) domain-independent information from the search process. It is worth mentioning, however, that low-level heuristics which encapsulate domain-specific information can be (and usually are) incorporated in the pool of available heuristics.

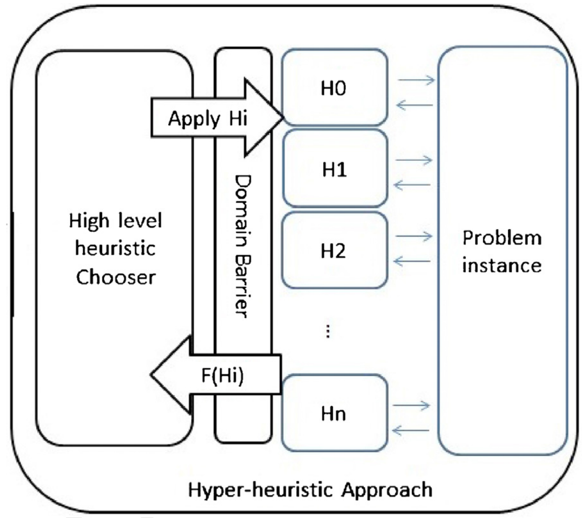

Fig. 1. General framework of a selection hyper-heuristic based on Cowling et al. [3].

When a hyper-heuristic uses some feedback from the search process, it can be considered as a learning algorithm (Fig. 1). According to the source of the feedback during learning, Burke et al. [4] distinguishes between *online* and *offline* learning hyper-heuristics, i.e. online learning takes place whilst a given algorithm is solving a problem instance.

In offline learning, the idea is to gather knowledge (e.g. in the form of rules or programs), from a set of training instances, in expectation of generalizing to unseen instances. Genetic Programming is one of the most commonly used methods for heuristic generation. Examples of off-line heuristic generation include [23,24] with [25] introducing a policy-matrix representation to inform the generation of heuristics. The add–delete hyper-heuristic proposed in this study is a novel online heuristic-generation method.

2.2. Educational timetabling

Although it has been extensively studied, educational timetabling problems are still of interest to many researchers and practitioners. There are many types of educational timetabling problems and this section focuses on a specific type of educational timetabling problem, that is, university course timetabling, in which the main objective is to assign each subject a timeslot such that that they attend all lectures to which they are enrolled. Formally, the university course timetabling problem can be considered as a Constraint Satisfaction Problem (CSP) where the variables are events and the most common constraints are time-related. A more detailed explanation of each timetabling variant used in this paper can be found in Section 2.2.1. This problem is reported as extremely challenging by Cooper et al. [6] and Willmen et al. [7].

Many approaches have been proposed for solving variants of educational time-tabling problems, ranging from early approaches based on graph heuristics [26], linear programming [27] and logic programming [28,29] to metaheuristics including tabu search [30], genetic algorithms [31,32], ant colony optimization [33,34], variable neighborhood search [35], simulated annealing [36], among others. Various CSP solvers have also been proposed to solve timetabling problems [37,38]. In recent years, hyper-heuristics have been applied to timetabling with encouraging results [39–41]. A chronological order of the state of the art in educational timetabling can be seen in Table 1.

Please cite this article in press as: J.A. Soria-Alcaraz, et al., Iterated local search using an add and delete hyper-heuristic for university course timetabling, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.11.043